MULTI-DISK CATALOG

Sensible Software

MULTI-DISK CATALOG III

MDC III is a very fast, machine-language database program designed specifically for keeping track of the contents of your APPLE diskette library. MDC III requires only seconds to read **FILENAMES, FILETYPES, FILESIZES,** number of **FREE SECTORS** remaining on diskette, and actual **VOLUME NUMBER** from each of your diskettes. Both sides of a diskette can be loaded and assigned to the same DISK ID#. MDC III supports use of a two-character **CLASSIFICATION** field that can be used to group games, utilities, and other types of related files together. **TITLING** is supported to allow cataloging of PASCAL, FORTRAN, and CP/M diskettes. MDC III supports a fast Shell-Metzner sort on any of the five database fields. A unique "**LIST MASK**" provides a powerful search capability for interrogating the database for specific information. A "**FLIP DOS**" command allows MDC III to read directories from DOS 3.1, 3.2, and 3.3 disks and to store the resulting database on either a 13 or 16 sector diskette.

Hardware:	APPLE II or APPLE II PLUS One or two disk drives
Memory:	48K
Language:	6502 machine language
DOS:	3.1, 3.2, 3.2.1, or 3.3
Includes:	Diskette & User Manual
Price:	$25.00

Sensible Software

TABLE OF CONTENTS

INTRODUCTION

MULTI-DISK CATALOG III is a fast, machine-language program that reads catalogs on 13 and 16 sector DOS disks and stores the filenames from these catalogs into a database in RAM memory. Each RAM database can hold up to 950 filenames and can be stored off-line as a numbered "master" file (MF1 to MF9). MDC III can be used to track an unlimited number of filenames that are broken into "groups" of 950 names.

One of the major benefits that MDC III offers is the ability to rapidly locate files on any version DOS diskette. An additional feature is the ability to identify duplicate files that are wasting valuable disk space.

COMMANDS

o **ESCAPE TO MENU**

The ESCape or the CTRL/C keys can be used to cancel all commands in MDC III except the SEARCH MASK.

o **ADD/REPLACE DISK (ID# ___)**

This command is used to read filenames from the user's disk into the RAM database. All of the filenames from one side of a disk will be assigned to the specified DISK ID number.

Note: This command first DELETES any filenames from the RAM database that were already assigned to the specified DISK ID number. However, MDC III will remember the old "classification"'s from the deleted files and will copy them to the new files being loaded.

Note: DISK ID numbers do not have to match disk volume numbers. It is suggested that the user apply a small, gummed label to the disk to show the DISK ID number that he has chosen.

Note: DISK ID numbers do not have to be consecutive. However, storing disks by sequential DISK ID number will make it easier to physically locate disks later.

o **2ND SIDE OF DISK**

This command allows the user to assign the filenames from the back side of a disk to the same DISK ID number that was used for the front side of the disk. This command automatically uses the last DISK ID number entered for the add/replace command.

Note: Do not use this command if you want to be able to later search for filenames on specific sides of disks -- use a different DISK ID number for the back side (for example, odd numbers for fronts, even for backs).

o CLASSIFY NAMES

This command lets you assign or change the 2 letter "classification" code for filenames in the RAM database. This command uses the "search mask" (explained below) to select filenames to be classified. A menu of suggested classifications will be displayed to assist you in classifying your files. This internal menu is intended to serve only as a guideline, the user can use any 2 letter codes of his own choice for classifying files.

Note: The built-in "guideline" menu cannot be changed by the user.

o DELETE DISK (ID# ___)

This command removes all filenames with the corresponding DISK ID number from the RAM database.

Note: Deleted names don't completely go away until the RAM database is sorted.

o EXIT

This command is equivalent to a PR#6 command except that the user is given an opportunity to swap disks before the new disk is "boot"'ed.

o FLIP DOS VERSIONS

This command toggles between 13-sector DOS (DOS 3.1, 3.2, & 3.2.1) and 16-sector DOS (DOS 3.3). The present setting is always shown in the upper right-hand corner of the screen.

Note: The proper DOS must be selected prior to any disk activity or else an "I/O ERROR" will result.

o GET DATABASE (# ___)

This command will load a database from disk back into RAM memory (assuming that the database has already been "saved"). This command will first display a catalog to remind the user what master file number ("MF 1" to "MF 9") was used to store the database. The user will then be prompted for the number portion of the master file name (for example, load "MF 1" by typing the number "1").

o LIST (W/SEARCH MASK)

This command lists the filenames in the RAM database in a columnar format. The search mask (explained below) can be used to limit the display to specified filenames.

Note: Use the LIST command to "find" files.

Note: The paddle controller can be used to control the speed of the listing.

Note: Press any key during the listing to "pause". Press a key again to "resume".

Note: ESCape or CTRL/C can be used to abort the listing.

o NEW DATABASE

This command deletes all names from the RAM database.

o ORDER NAMES (&PACK DATABASE)

This command lets you select one, two, or three of the five columns from the LIST report to be sorted. A sub-menu showing the names of the five LIST columns will appear, followed by the question "ENTER SORT FIELDS?". Enter the first letters of the name of the LIST columns that you want sorted. For example, if you just want to sort by filename, enter an "F" and then press "RETURN". If you want to sort by filename within DISK ID number, enter "I", then enter "F", and then press "RETURN".

Note: Fields are always sorted in ascending order.

Note: If three LIST columns (or fields) are specified for the sort, you do not have to press "RETURN".

Note: The speaker will be "clicked" whenever two names are swapped during the sorting process.

Note: Sorting removes "deleted" names from the RAM database, giving more room to store new filenames.

o PR# (# ___)

This command allows you to enable the printer for the next LIST command. Enter a "1" if an APPLE serial card is in slot 1 and the output from the next LIST command will go to the printer.

Note: This command must be issued before each LIST that is to be printed. A PR#0 is automatically issued at the end of each LIST!

o QUIET THE SORT NOISE

This command removes the speaker "clicking" during the sort process!

o SAVE DATABASE (# ___)

This command will save a RAM database to a binary disk file. This command will first display a catalog to remind the user what master file numbers ("MF 1" to "MF 9") have already been used. The user will then be prompted for the number portion of the master file name (for example, save "MF 1" by typing the number "1"). Most users (having less than 950 files) will only need to specify "MF 1" for all of their work.

o TITLE DISK (ID# ___)

This command allows the user to add a short descriptive title to a DISK ID number. This command is independent of all other commands except "DELETE DISK ID" and "NEW DATABASE". It is not necessary to load any filenames or to read any disks to be able to use this command, thus this command can be used to add titles to help locate non-DOS disks (PASCAL, FORTRAN, PILOT, CP/M, etc.)

Note: Only one title is allowed per DISK ID number.

THE SEARCH MASK CONCEPT

MDC III uses a "mask" instead of the "traditional" sub-string search. The sub-string search is most useful when you are only searching one field. For example, it would be handy to find all filenames with "APPLE" in them in the FILENAME field. The problem starts getting more complex when you are dealing with 5 fields (as in MDC III) and you want to be able to ask questions about any combination of the five fields. For example, find all filenames that have "APPLE" in the FILENAME field and that have "B" in the TYPE field and that are larger than 100 sectors in the SIZE field.

The search "mask" provides an easy way to ask questions about any combination of the five database fields. The "mask" is the line that you enter in response to the "ENTER SEARCH MASK" question. Any non-blank characters in the "mask" will be compared against each name in the RAM database. Only the entries in the RAM database that exactly match every non-blank letter in your "mask" are going to be "found".

The key is that the "mask" works column by column.

The disadvantage of the "mask" compared to the "sub-string" search is that a substring search for "APPLE" would find both "APPLESOFT" and "MY APPLE" while a mask search for "APPLE" would find only "APPLESOFT".

Examples:

(in the following database:)

```
ID#  CL   T    SIZ  FILENAME

.10  ..   B    .58  MY FILE #1
.80  DU   I    ..2  FULL DISK CATALOG
110  ..   A    123  MY FILE
211  DU   I    ..3  DISK COPY PROGRAM
211  DU   I    ..8  DISK MAP
```

```
ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

?<RETURN>

 (would find:)
.10  ..   B    .58  MY FILE #1
.80  DU   I    ..2  FULL DISK CATALOG
110  ..   A    123  MY FILE
211  DU   I    ..3  DISK COPY PROGRAM
211  DU   I    ..8  DISK MAP
```

```
ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

?                    MY FILE

(would find:)
.10  ..   B    .58  MY FILE #1
110  ..   A    123  MY FILE


ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

?          A         MY FILE

(would find:)
110  ..   A    123  MY FILE


ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

?    DU

(would find:)
.80  DU   I    ..2  FULL DISK CATALOG
211  DU   I    ..3  DISK COPY PROGRAM
211  DU   I    ..8  DISK MAP


ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

? 1

(would find:)
.10  ..   B    .58  MY FILE #1
110  ..   A    123  MY FILE
211  DU   I    ..3  DISK COPY PROGRAM
211  DU   I    ..8  DISK MAP
```

```
ENTER SEARCH MASK

ID#  CL   T    SIZ  FILENAME

?:1

(would find:)
.10  ..   B    .58  MY FILE #1
```

The colon (in the above example) is necessary to avoid matching on DISK ID numbers "211" and "110" since they also have a "1" in the second column! The colon only matches on a blank in the corresponding column.

PRINTER INTERFACE CUSTOMIZATION

If you are one of those frugal, hardy souls that invested in a "non-standard" printer interface (such as the game I/O-connector RS-232 driver from the APPLE "red" book), you may still be able to get MDC III to talk to your printer.

If you type a "CTRL/P" instead of a slot number in response to the "PR# ()" question, MDC III will call your print routine with a "JSR $9200" each time it wants to print a character. The character to be printed will be passed as ASCII in the "A" register with the high-order bit set. It is the responsibility of your printer subroutine to return the "X", "Y", and stack registers the way they were given to you!

Now to make matters a little more complicated, your routine must fit in locations $9200 to $9250. Your routine can't just be loaded into location $9200 directly either, because of the way that MDC III "boots". You must load it into memory locations $4000-$4050 prior to "boot"ing MDC III and then MDC III will copy it up to $9200-$9250. This can be taken care of with the sample loader program shown below. A sample (very simple) printer driver for the Electronic Systems "dumb" serial I/O board is also shown.

MODIFIED PRINTER ROUTINE

```
                5               ORG   $9200
                6               OBJ   $4000
                9                                 ;SEND A-REG. TO SER OUT + VIDEO
     20F0FD     10              JSR   VIDEO
     0980       11              ORA   80
     C98D       12              CMP   #RETURN     ;IF <CR>
     D007       13              BNE   SNDCHR      ; THEN
     A98A       14              LDA   #LNFEED     ;DO <CR/LF>!
     201092     15              JSR   SNDCHR
     A98D       16              LDA   #RETURN
                17                                ;SEND CHAR CODE FOR ELEC SYS
                18    SNDCHR:
     6E81C0     19              ROR   STATUS      ;CLEAR TO SEND?
     90FB       20              BCC   SNDCHR
     8D82C0     21              STA   SEROUT      ;SEND CHAR.
     60         22              RTS
                23                                ;
                24    VIDEO     EQU   $FDF0
                25    RETURN    EQU   $8D         ;<CR>
                26    LNFEED    EQU   $8A         ;<LF>
                27    STATUS    EQU   $C0B1
                28    SEROUT    EQU   $C0B2
                29                                ;
                30    END=      END
```

```
BSAVE MDC.PRINT,A$4000,L$50
```

(boot a "normal" DOS 3.2 disk and then run the following program on the MDC III disk:)

```
100 D$=""                                  : REM CTRL/D IN QUOTES
200 PRINT D$;"BLOAD MDC.PRINT,A$4000"
300 INPUT"INSERT MDC DISK & PRESS RETURN",A$
400 PRINT D$;"PR#6"
500 END
```

SAMPLE SESSION (user responses in boldface)

(insert MDC III disk)

>pr#6

(menu appears showing 0 files loaded)

ENTER COMMAND? **n** (new database)

(menu appears showing 0 files loaded)

(remove MDC III disk, select one of your DOS 3.2 disks and assign it a disk id #, let's use 122 for this example. Place a gummed label on the disk with the number 122 on it, then insert the disk into drive 1)

ENTER COMMAND? **a** (add disk)

ADD DISK ID #? **122** (your disk id#)

(disk spins momentarily)

(menu appears showing 14 files loaded)

(let's say that you want the files on the back of that disk to be assigned to disk id# 123. Remove the disk, add a second gummed label with the number 123 and insert the reverse side into the drive)

ENTER COMMAND? **a** (add disk)

ADD DISK ID #? **123** (your disk id#)

(disk spins momentarily)

(menu appears showing 31 files loaded)

(select one of your DOS 3.3 disks and assign it a disk id of 47. Place a gummed label with the number 47 on it and place it into drive 1)

ENTER COMMAND? **f** (flip dos)

(menu appears showing DOS 3.3)

ENTER COMMAND? **a** (add disk)

ADD DISK ID #? **47** (your disk id#)

(disk spins momentarily)

(menu appears showing 43 files loaded)

(let's say that you have more files on the back of that disk that you also want to assign to disk id# 47. Remove the disk and insert the reverse side)

ENTER COMMAND? 2 (add 2nd side)

(disk spins momentarily)

(menu appears showing 54 files loaded)

(ok, let's save your work so far -- insert a normally initialized disk and continue with this demo. The back side of the MDC disk can be used if a notch is cut in the left side of the disk across from the notch on the right side.)

ENTER COMMAND? s (save database)

(catalog appears, press space bar to continue)

SAVE MASTER FILE #? 1 ("MF 1")

(light flashes in bottom right corner for a moment, then disk whirs for a while. Finally, menu reappears -- you can shut off your APPLE now)

(you return the next day to continue)

(insert MDC III disk)

>pr#6

(menu appears showing 0 files -- insert the disk that you saved the data file "MF 1" onto and continue with the demo.)

ENTER COMMAND? g (get database)

(catalog appears, note the "MF 1" file that has the names that you previously saved. Press space bar to continue)

GET MASTER FILE #? 1 ("MF 1")

(disk whirs for a while, then light flashes in lower right corner for a moment. Finally, menu reappears showing 52 files loaded)

(now you want to see what you have already loaded by "listing" the database)

ENTER COMMAND? l (list database)

ENTER SEARCH MASK BELOW

ID# CL T SIZ FILENAME

?<return>

(you press the "RETURN" key when the question mark appears so that all of the files in the database will be listed. The names then appear on the screen, press the space-bar to continue)
(the menu reappears after you press the space-bar)

(you now want to add a third disk to the database that you are numbering as disk id# 667. Remove the MDC III disk and insert the new disk)

ENTER COMMAND? a (add disk)

ADD DISK ID #? **667** (your disk id)

(the disk whirs a moment, then the menu reappears showing 73 files loaded. At this point, you might want to resave the database as "MF 1" with the new information!)

(let's say that you now want to find the file "MY PROGRAM". This is done by "listing" the file as follows:)

ENTER COMMAND? l (list database)

ENTER SEARCH MASK

ID# CL T SIZ FILENAME

? **my file**

(press the space bar until the cursor is under the "F" of filename and type the filename, then press "RETURN". All of the files in the database whose names start with "my file" will then be listed. The number in the far left column tells you which disk the program is located on. Let's say that the program is on disk 47)

(remove the MDC III disk and insert the disk with the disk id# 47)

ENTER COMMAND? e (exit MDC!)

INSERT BASICS DISK AND PRESS RETURN

(pressing return boots the disk with "my program" and your off and running!)

ACCESS TO MF FILES FROM BASIC

The following Applesoft program listing is presented only as an example of how to access and manipulate the database generated by MDC III. By using this example and your own ingenuity you are no longer limited to the commands offered by MDC III. Printer listings of your diskette library can be formatted specifically to your own needs. We realize that access to the MF files can be accomplished in a more sophisticated manner but offer this listing as an example only.

```
10   REM  [ MDC TWEAKER.APLUS ]
20   REM
30   REM  (C) 1981 - SENSIBLE SOFTWARE
40   REM  ALL RIGHTS RESERVED
50   REM
70   REM  allocate memory
80   REM
90   REM   NOTE: following memory
100  REM  limits must not be exceeded
110  REM  (this prevents the use of RAM
120  REM  applesoft.)
130  REM
140  INC = 32:FIRST = 8448: REM $2100
150  PRINT  CHR$ (4)"MAXFILES 1"
160  HIMEM: FIRST
170  REM
190  REM  select & load a master file
200  REM
210  INPUT "INSERT MDC DATA DISK";A$
220  PRINT  CHR$ (4)"MON CIO"
230  PRINT  CHR$ (4)"CATALOG,D1"
240  PRINT
250  INPUT "LOAD MASTER FILE (MF #)? ";MF
260  PRINT  CHR$ (4)"BLOAD MF "MF", A"FIRST
265  REM =(A$...+L$...)
270  LAST = FIRST +  PEEK (43616) + 256 *  PEEK (43617)
280  REM
290  REM
300  REM  merge with a 2nd "mf #"?
310  REM
320  PRINT
330  INPUT "MERGE WITH MF# (0=NONE)? ";M2
340  IF  NOT (M2) GOTO 370
350  PRINT  CHR$ (4)"BLOAD MF "M2", A"LAST
355  REM =(A$...+L$...)
360  LAST = LAST +  PEEK (43616) + 256 *  PEEK (43617)
370  REM ::
380  REM
```

```
390  REM
400  REM  convert the "inverse" ascii
410  REM  to "normal" applesoft ascii
420  REM
430  REM CONVERT TO POSITIVE ASCII
440  DEF  FN P(B) = B - 128 *  INT (B / 128)
450  REM REMOVE CONTROL CHARACTERS
460  REM IFB<$20THENB=B+$40
470  DEF  FN R(B) = B + 64 * (1 -  SGN ( INT (B / 32)))
480  REM GET "CLEAN" ASCII
490  DEF  FN V(B) =  FN R( FN P(B))
500  REM PEEK "CLEAN" ASCII BYTE
510  DEF  FN A(ADDR) =  FN V( PEEK (ADDR))
520  REM
530  REM
540  REM  delete files with names
550  REM  that fall outside of a
560  REM  specified range?
570  REM
580  PRINT
590  INPUT "DO YOU WANT TO DELETE FILENAMES OUTSIDE OF A SPECIFIED
     RANGE (Y/N)?";A$
600  IF  NOT (A$ = "Y") GOTO 790
610  PRINT
620  PRINT "KEEP FILES WHOSE NAMES"
630  PRINT "BEGIN WITH THE LETTERS:"
640  INPUT "   FROM THE LETTER?";FIRST$
650  INPUT "THROUGH THE LETTER?";LAST$
660  I = FIRST:NUM = 1: PRINT : PRINT
670  REM NTIL((I >  = LAST) OR (ID = 65535))
680  VTAB 23: PRINT "FILE# ";NUM;" ";
690  ID =  PEEK (I + 1) + 256 *  PEEK (I)
700  CHAR$ =  CHR$ ( FN A(I + 6)): REM 1ST LETTER IN NAME$
710  IF (FIRST$ <  = CHAR$ AND CHAR$ <  = LAST$) GOTO 760
720  POKE (I + 4),90: REM 'Z'=DELETE FLAG
730  INVERSE
740  PRINT "DELETED"
750  NORMAL
760  REM ::
770  I = I + INC:NUM = NUM + 1: PRINT
780  IF  NOT ((I >  = LAST) OR (ID = 65535)) GOTO 670
790  REM ::
800  REM
```

```
810  REM
820  REM  display the remaining names
830  REM  from RAM memory.
840  REM
850  I = FIRST
860  REM UNTIL((I >  = LAST) OR (ID = 65535))
870  ID =  PEEK (I) + 256 *  PEEK (I + 1)
880  CLASS$ =  CHR$ ( FN A(I + 2)) +  CHR$ ( FN A(I + 3))
890  TYPE$ =  CHR$ ( FN A(I + 4))
900  IF (TYPE$ = "Z") GOTO 1270
910  SIZE =  PEEK (I + 5)
920  NAME$ = ""
930  FOR COLUMN = 6 TO INC - 1
940  NAME$ = NAME$ +  CHR$ ( FN A(I + COLUMN))
950  NEXT COLUMN
960  REM ASE
970  IF  NOT (ID < 10) GOTO 1000
980  PRINT "  ";ID;
990  ::: GOTO 1060
1000  IF  NOT (ID < 100) GOTO 1030
1010  PRINT " ";ID;
1020  ::: GOTO 1060
1030  REM ::::::::::
1040  PRINT ID;
1050  REM ::
1060  REM ::
1070  PRINT " ";CLASS$;
1080  PRINT " ";TYPE$;
1090  PRINT " ";
1100  REM ASE
1110  IF  NOT (SIZE = 0) GOTO 1160
1120  INVERSE
1130  PRINT "-->";
1140  NORMAL
1150  ::: GOTO 1250
1160  IF  NOT (SIZE < 10) GOTO 1190
1170  PRINT "  ";SIZE;
1180  ::: GOTO 1250
1190  IF  NOT (SIZE < 100) GOTO 1220
1200  PRINT " ";SIZE;
1210  ::: GOTO 1250
1220  REM ::::::::::
1230  PRINT SIZE;
1260  PRINT " ";NAME$
1280  I = I + INC
1290  IF  NOT ((I >  = LAST) OR (ID = 65535)) GOTO 860
1300  REM
1310  REM
1320  REM  save modified master file
1330  REM
1340  PRINT  CHR$ (4)"BSAVE NEW MF, A"FIRST", L"LAST - FIRST
1350  PRINT
1360  PRINT  CHR$ (4)"CATALOG"
1370  END
```

THE "MENU" PAGE

```
-----------------------------------------
| MULTI-DISK  CATALOG  653 NAMES/DOS3.3
| COPYRIGHT  (C) 1980
| ALL RIGHTS RESERVED-SENSIBLE SOFTWARE-
|        your name here .......
|
| ESCAPE TO MENU
| ADD/REPLACE DISK
| 2ND SIDE OF DISK
| CLASSIFY NAMES
| DELETE DISK
| EXIT
| FLIP DOS VERSION
| GET DATABASE
| LIST
| NEW DATABASE
| ORDER NAMES (&PACK DATABASE)
| PR#
| QUIET THE SORT NOISE
| SAVE DATABASE
| TITLE DISK
|
| ENTER 1ST LETTER OF COMMAND ?
-----------------------------------------
```

DISCLAIMER

Sensible Software, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantibility or fitness for any particular purpose. Further, Sensible Software, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Sensible Software, Inc. to notify any person or organization of such revisions or changes.

DISK REPLACEMENT/UPDATE POLICY

Some Sensible Software products are non-copyable. In the event that one of these "protected" disks becomes damaged or "worn out", the following policy applies:

Upon receipt of the (remains of the) disk that was originally purchased from Sensible Software and the appropriate fee (see below), Sensible Software will return to the purchaser a disk containing the latest version of the original program product. Sensible Software reserves the right to return the program product on a diskette other than the one originally purchased. If the product is still available for the retail price in effect at the time of the original purchase, the fee for this update/replacement service is $5.00 to cover postage and handling. If the program product now retails at a higher price, the fee for this update/replacement service is $2.00 plus the difference between the two retail prices.

Sensible Software
6619 Perham Drive
West Bloomfield, Michigan 48033
(313) 399-8877